IN THE SPECIFICATION:

Please replace the text on page 5, lines 9-10, with the following rewritten text:

A1

Figures 2A and **2B** are block diagrams illustrating ~~example~~ examples of shared cache read sequences in accordance with a preferred embodiment of the present invention;

Please replace the text on page 5, lines 9-10, with the following rewritten text:

A2

Figures 3A and **3B** are block diagrams illustrating ~~example~~ examples of shared cache write sequences in accordance with a preferred embodiment of the present invention;

Please replace the paragraph on page 6, lines 3-9, with the following rewritten paragraph:

A3

The description of the preferred embodiment of the present invention has been presented for purposes of illustration and description, but is not ~~limited~~ intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention and the practical application to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

Please replace the paragraph on page 6, lines 10-19, with the following rewritten paragraph:

A4

With reference now to the figures and in particular with reference to **Figure 1**, a block diagram of a switched controller architecture is depicted in accordance with a preferred embodiment of the present invention. The architecture includes a first controller **100** and a second controller **150** to provide full path redundancy to host computer systems. Controller **100** includes host channel adapters (CA) **102**, **104** and drive channel adapters **106**, **108**. The host channel adapters are the physical connections between the internal bus and the host interface. The internal bus may be, for example, an

~~Infiniband~~ INFINIBAND bus. While the example shown in **Figure 1** is an ~~Infiniband~~ INFINIBAND architecture, controllers **100, 150** may be any other switched architecture. "INFINIBAND" is a trademark of System I/O, Inc., DBA InfiniBand Trade Association, Portland, Oregon. The drive channel adapters are the physical connections between the internal bus and the drive interface.

A4

Please replace the paragraph on page 6, lines 20-24, with the following rewritten paragraph:

Controller **100** also includes central processor unit (CPU) **110**. The CPU may have an associated random access memory (RAM) **112** as a working memory. Further, controller **100** includes remote memory controllers (RMC) **122, 124**. [[A]] An RMC is the control hardware for managing the connection to a memory. RMC **122** manages the connection to RAM **126** and RMC **124** manages the connection to RAM **128**.

A5

Please replace the paragraph on page 8, line 7, to page 9, line 2, with the following rewritten paragraph:

With reference now to **Figures 2A** and **2B**, block diagrams illustrating ~~example~~ examples of shared cache read sequences are shown in accordance with a preferred embodiment of the present invention. Particularly, with respect to **Figure 2A**, a read request is processed according to the following steps:

1. A read request is received by Controller A.
2. Controller A allocates memory buffers for the read data. Because the logical cache memory pool resides on both controllers, the memory buffer could be allocated from either controller's physical memory pool. In the example shown in **Figure 2A**, the buffer for the read request received by Controller A happens to be on Controller B. By definition, however, this buffer could be allocated on either controller. It is during this allocation phase that cache coherency must be maintained between the controllers. The LBA extent for this read is marked as locked, such that other reads to the same LBA on either controller is forced to wait for the disk read to complete for the read received by Controller A. Once the memory buffer is allocated, Controller A maps the

A6

request to the appropriate disk drives and initiates reads (data transfers) via the appropriate drive CA from the disk drives. The reads do not necessarily have to occur through the drive CA on the controller that received the original read request.

3. The drive CA begins to transfer the data to the appropriate memory pool. This step facilitates future cache read hits for this data. Because of the concurrent cache coherency management inherent in this approach, subsequent reads of the same LBA to either controller would discover the data in the logical cache pool.

A6

4. Data is transferred to the host CA on Controller A that received the request.
5. Controller A directs command status to be returned through the originating CA on that controller.

Please replace the paragraph on page 9, lines 12-27, with the following rewritten paragraph:

With reference now to **Figures 3A** and **3B**, block diagrams illustrating ~~example~~ examples of shared cache write sequences are shown in accordance with a preferred embodiment of the present invention. Particularly, with respect to **Figure 3A**, a write request is processed with write-back caching according to the following steps:

1. A write request is received by Controller A. Controller A allocates memory buffers for the request. Two buffers are allocated, one on Controller A and another on Controller B. These two buffers serve as mirrors of each other. In order to maintain cache coherency, the LBA extent is locked to prevent other access to this data by requests received by either controller.

A7

2. The data transfer is initiated by Controller A. The originating host CA begins to transfer data to the primary data buffer via the appropriate RMC. Although the example in **Figure 3** shows that the primary data buffer resides on Controller A, the primary data buffer may reside on either controller. However, it is required that the mirror buffer reside on the controller that does not contain the primary data buffer.

A7

3. Data is transferred to the appropriate RMC and data buffer on the alternate controller.
4. Controller A directs command status to be returned through the originating CA.

Please replace the paragraph on page 10, line 27, to page 11, line 6, with the following rewritten paragraph:

A8

Turning now to **Figure 5**, a flowchart is shown illustrating the processing of a write request ~~is shown~~ in accordance with a preferred embodiment of the present invention. The process begins when a write request is received and the process allocates memory buffers for write data (step **504**) and transfers data to the primary data buffer (step **506**). Thereafter, the process transfers data to the mirror data buffer (step **508**). The primary buffer need not reside on the controller that receives the write request. However, the mirror buffer must reside on a controller, which does not contain the primary buffer to avoid a single point of failure. Next, the process returns command status (step **510**) and ends. Alternatively, if the volume is configured for write-through caching, the controller directs write completion to the disk drives before returning status in step **510**.
